



Universidade do Minho

Mestrado Integrado em Engenharia Informática
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Lectivo de 2020/2021

Uminho Entregas

André Araújo - a87987

Carlos Ferreira - a87953

Daniel Ribeiro - a87994

Paulo Costa - a87986

Novembro, 2020

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

Uminho Entregas

André Araújo - a87987

Carlos Ferreira - a87953

Daniel Ribeiro - a87994

Paulo Costa - a87986

Novembro, 2020

<</opcional Dedicatória>>

Resumo

O seguinte relatório documenta, justifica, analisa e expõe todas as decisões tomadas ao longo do projeto "UminhoEntregas" realizado no âmbito da Unidade Curricular denominada Bases de dados no contexto do 3º ano do curso universitário Licenciatura em Ciências da Computação.

Ao longo deste relatório vamos apresentar as nossas ideias de forma clara e objetiva, apoiando a sua explicação através de esquemas e tabelas para uma melhor compreensão por parte do leitor.

No início é feito uma introdução e contextualização do problema, apresentamos assim a necessidade de uma base de dados para esta empresa e a sua viabilidade.

Após isso, fazemos a análise de requisitos, onde determinamos os requisitos de descrição, exploração e controlo necessários para o projeto.

No terceiro capítulo, procedemos á modelação conceitual do projeto com base nos requisitos anteriores, apresentando todas as entidades, atributos e relações necessárias.

No quarto capítulo, transitamos do modelo conceitual para o lógico detalhando e aprofundando todas as entidades e relacionamentos seguindo todas as regras da normalização, validando assim o modelo.

No quinto capítulo é feita a passagem deste modelo lógico para o físico, gerando as instruções SQL para criar a base de dados, fazemos também um povoamento da mesma e alguns exemplos de "Queries" que podem ser usadas para exploração da mesma, fazemos uma estimativa do espaço em disco ocupado pela nossa base de dados e a sua taxa de crescimento anual também uma análise das vistas de utilização e acabamos com uma revisão da mesma.

Por fim, concluímos o trabalho realizado, com uma apreciação crítica do mesmo, especificando os seus pontos fortes e fracos analisando também a escalabilidade e extensão do mesmo.

Área de Aplicação: Desenho e Arquitetura de Sistemas de Base de Dados para uma empresa de transportes.

Palavras-Chave: SQL, Entregas, Funcionarios, Clientes, Veiculos, Base de Dados, Base de Dados Relacionais, Levantamento e análise de Requisitos, Modelo Conceptual, Modelo Lógico, Modelo Físico, Entidades, Atributos, Relacionamentos, Query, brModelo, MySQL Workbench,

Índice

Resumo	1
Índice de Figuras	5
Índice de Tabelas	7
1. Definição do Sistema	8
1.1 Contexto de aplicação do sistema	8
1.2 Fundamentação da implementação da base de dados	8
1.3 Análise da viabilidade do processo	9
2. Levantamento e Análise de Requisitos	10
2.1 Método de levantamento e de análise de requisitos adotado	10
2.2 Requisitos levantados	10
2.2.1 Requisitos de descrição	10
2.2.2 Requisitos de exploração	11
2.2.3 Requisitos de controlo	11
2.3 Análise e validação geral dos requisitos	12
3. Modelação Conceptual	13
3.1 Apresentação da abordagem de modelação realizada	13
3.2 Identificação e caracterização das entidades	14
3.3 Identificação e caracterização dos relacionamentos	15
3.4 Identificação e caracterização da associação dos atributos com as entidades e relacionamentos.	15
3.5 Detalhe ou generalização de entidades	18
3.6 Apresentação e explicação do diagrama ER	19
3.7 Validação do modelo de dados produzido	20
4. Modelação Lógica	21
4.1 Construção e validação do modelo de dados lógico	21
4.1.1 Derivação das relações	21
4.1.2 Derivação dos relacionamentos	22
4.1.3 Análise das restrições de integridade	23
4.2 Desenho do modelo lógico	25
4.3 Validação do modelo através da normalização	26
4.4 Validação do modelo com interrogações do utilizador	26
4.5 Revisão do modelo lógico produzido	27
5. Implementação Física	28
5.1 Seleção do sistema de gestão de bases de dados	28
5.2 Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL	29
5.3 Tradução das interrogações do utilizador para SQL (alguns exemplos)	36

5.4 Escolha, definição e caracterização de índices em SQL (alguns exemplos)	43
5.5 Estimativa do espaço em disco da base de dados e taxa de crescimento anual	44
5.6 Definição e caracterização das vistas de utilização em SQL (alguns exemplos)	47
5.7 Revisão do sistema implementado	48
6. Conclusões e Trabalho Futuro	49
7. Referências	50
Lista de Siglas e Acrónimos	51
Anexos	52
Anexo 1	53

Índice de Figuras

Figura 1 - Modelo Conceptual da UminhoEntregas	19
Figura 2 - Modelo Lógico da UminhoEntregas	25
Figura 3 - Criação do espaço de trabalho	29
Figura 4 - Criação da tabela Funcionario	29
Figura 5 - Criação da tabela Cliente	29
Figura 6 - Criação da tabela Encomenda	30
Figura 7 - Criação da tabela Veiculo	31
Figura 8 - Criação da tabela Contacto	31
Figura 9 - Criação da tabela Utiliza	32
Figura 10 – Função calculaValor para calcular o valor a pagar na fatura	32
Figura 11 – Função de inserção de um novo Cliente	33
Figura 12 – Função de inserção de um novo Contacto	33
Figura 13 – Função de inserção de uma nova Encomenda	33
Figura 14 – Função de inserção de um novo Funcionário	33
Figura 15 – Função de inserção na tabela utiliza	34
Figura 16 – Função de inserção de um novo veículo	34
Figura 17 – Povoar a tabela Clientes	34
Figura 18 – Povoar a tabela Funcionários	34
Figura 19 – Povoar a tabela Contactos	35
Figura 20 – Povoar a tabela Veículos	35
Figura 21 – Povoar a tabela Encomendas com encomendas	35
Figura 22 – Povoar a tabela utiliza	36
Figura 23 - Procedimento para visualizar Cliente	36
Figura 24 – Procedimento para visualizar Funcionarios	36
Figura 25 – Procedimento para visualizar Veiculos	37
Figura 26 – Procedimento para visualizar Encomendas	37
Figura 27 – Chamada das funções	37
Figura 28 – Funções executaram corretamente	37
Figura 29 – Resultado da visualizarClientes	37
Figura 30 – Resultado da visualizarFuncionarios	38

Figura 31 – Resultado da visualizarVeículos	38
Figura 32 – Resultado da visualizarEncomendas	38
Figura 33 – Query para 2º Requisito	39
Figura 34 – Chamada da função	39
Figura 35 – Encomendas do cliente com NIF 123456789	39
Figura 36 – Query para 3º Requisito	
Figura 37 – Chamada da função	39
Figura 38 – Encomendas entregues pelo funcionário 1 entre dia 28 e 30 de Outubro 2020	40
Figura 39 – Query para 4º Requisito	40
Figura 40 – Chamada da função	40
Figura 41 – Valor pendente do cliente	40
Figura 42 – Query para 5º Requisito	41
Figura 43 – Chamada da função	41
Figura 44 – Desempenho dos funcionários	41
Figura 45 – Query para 6º Requisito	41
Figura 46 – Chamada da função	41
Figura 47 – Utilização dos veículos	42
Figura 48 – Query para 7º Requisito	42
Figura 49 – Chamada da função	42
Figura 50 – Top 3 clientes	42
Figura 51 – Requisitos de Controlo	43
Figura 52 - Criação do índice no Estado da Encomenda	43
Figura 53 - Tabela para mostrar informação de fatura	47
Figura 54 - Tabela representativa de diferentes medias	47
Figura 55 - Tabela que mostra as encomendas em falta	48

Índice de Tabelas

Tabela 1 - Dicionário de dados de entidades	14
Tabela 2 - Dicionário sobre os dados dos relacionamentos	15
Tabela 3 - Dicionário sobre os atributos das Entidades	16-18
Tabela 4 - Cálculo do tamanho de cada registo do Funcionario	44
Tabela 5 - Cálculo do tamanho de cada registo do Cliente	44
Tabela 6 - Cálculo do tamanho de cada registo da Encomenda	45
Tabela 7 - Cálculo do tamanho de cada registo do Veiculo	45
Tabela 8 - Cálculo do tamanho de cada registo do Contacto	46
Tabela 9 - Cálculo do tamanho de cada registo do Utiliza	46

1. Definição do Sistema

1.1 Contexto de aplicação do sistema

A empresa Uminho Entregas, é uma empresa de entrega de encomendas que está localizada numa pequena localidade na região de Braga.

Esta pequena empresa é gerida pelo Sr.Manuel, uma pessoa já de idade que é responsável por receber os pedidos das pessoas, este guarda os pedidos no seu caderno e calcula o valor do serviço. Esses pedidos são depois distribuídos pelos seus funcionários que realizam as entregas.

Mas com o agravamento da doença Covid-19, as pessoas começaram a evitar sair de casa ao máximo e por isso a empresa Uminho Entregas estava a receber um grande número de pedidos, e o Sr.Manuel já não conseguia manter o ritmo, como resultado alguns dos pedidos estavam a ser mal anotados e por isso os pedidos não estavam a ser realizados com o sucesso que o Sr.Manuel pretendia.

Então este decidiu investir na implementação de uma base de dados que o permitiria armazenar e controlar facilmente a informação relativa aos serviços prestados pela empresa.

1.2 Fundamentação da implementação da base de dados

O crescimento exponencial de serviços prestados pela Uminho Entregas, e o facto de quererem permanecer como a principal empresa de transportes de encomendas da sua localidade, fez com que fosse uma prioridade a implementação de uma base de dados, que os iria permitir armazenar os registos referentes aos serviços prestados, de forma rápida e eficiente, de forma também a facilitar o processo de consulta e gestão dos dados, e por isso manterem-se um passo a frente da concorrência.

O sistema de base de dados desenvolvido deve suportar o armazenamento e a consulta da documentação da empresa, de modo a que seja possível conhecer a qualquer momento, de forma rápida e simples:

- As encomendas transportadas
- O funcionário que realizou o transporte
- Informação referente ao cliente
- Os pagamentos efetuados pelos clientes

1.3 Análise da viabilidade do processo

Os principais elementos que influenciam na decisão dos clientes na escolha de qual empresa entregará a sua encomenda são o estado com que chega a encomenda ao seu destino e o tempo que a encomenda demora a ser entregue.

A implementação da base de dados nada afetará no estado com que a encomenda chega ao destino, sendo isso da responsabilidade da Uminho Entregas, mas garantirá uma diminuição significativa no tempo de entrega, mais concretamente o tempo que passa desde a receção do pedido da entrega até ao momento em que o cliente recebe a encomenda na porta de sua casa.

Tendo em conta que previamente o gerente da Uminho Entregas tratava da alocação, consulta e distribuição dos pedidos de entregas no seu caderno, os tempos de entrega não eram tão rápidos quanto podiam, pois a procura de um nome ou de uma encomenda no caderno podia-se tornar muito demorada, e os funcionários da empresa apenas teriam contacto com a informação dos pedidos depois de se dirigirem ao escritório do gerente e verificarem no caderno.

A base de dados permitirá que no futuro, o tempo de alocação e consulta de pedidos seja praticamente instantâneo, todos os funcionários terão acesso à versão atualizada da base de dados a todo o tempo, permitindo-os concluir pedidos muito mais rapidamente. Assim o tempo médio de entrega será reduzido significativamente e colocará a Uminho Encomendas como a candidata de escolha dos habitantes da região de Braga.

2. Levantamento e Análise de Requisitos

Uma das fases mais importantes de cada projeto é a análise de requisitos, os requisitos bem levantados e estruturados são a base para um bom projeto, esta deve ser feita de forma cuidadosa e organizada de forma a garantir todas as funcionalidades desejadas, desta forma o objetivo deste tópico será uma análise de todos os requisitos necessários para o bom funcionamento deste projeto.

2.1 Método de levantamento e de análise de requisitos adotado

Após uma análise de tudo o que foi pedido pelo Sr. Manuel, o grupo de trabalho começou a proposta de requisitos, sendo esta o resultado de extensa deliberação e análise crítica, de forma a alcançar uma solução correta e eficiente para o caso de estudo em questão.

2.2 Requisitos levantados

Iniciamos então o levantamento de requisitos divididos em 3 categorias, os requisitos de descrição, onde apresentamos os requisitos necessários para o funcionamento e armazenamento dos dados, os de exploração onde analisamos os requisitos necessários para apresentação e listagem de dados e os requisitos de controlo visando a proteção e encapsulamento de informações com diferentes modos de acesso á base de dados.

2.2.1 Requisitos de descrição

- O sistema deverá guardar as Encomendas realizadas, bem como o seu estado (Pedida, Paga, A caminho ou Entregue), data em que o cliente pediu a mesma, a data de entrega, a taxa (valor aplicado pela UminhoEntregas na encomenda), o peso, o local onde será entregue, o tipo de pagamento e se a mesma é urgente.

- Cada Encomenda têm uma fatura associada com o identificador da mesma, a data de emissão e valor.
- Cada cliente da empresa também deverá ser guardado na base de dados, assim como os dados relativos a ele o NIF, o Nome, o Email, morada e Contacto
- Cada cliente pode ter mais do que um contacto.
- Os Funcionários da UminhoEntregas também deverão ser guardados na base de dados junto com o seu nome e telemóvel, tendo cada funcionário um identificador único.
- Os veículos da empresa também têm de ser guardados na base de dados, identificados pela sua matrícula e com dados relevantes do mesmo, como a marca, modelo e ano de fabrico.

2.2.2 Requisitos de exploração

- Apresentar todos os clientes, funcionários, veículos ou encomendas.
- Apresentar as encomendas de um dado cliente.
- Apresentar as encomendas entregues num dado intervalo de tempo por certo funcionário.
- Apresentar o valor pendente de um dado cliente.
- Apresentar o desempenho dos funcionários num determinado dia.
- Apresentar os veículos mais utilizados pela empresa.
- Apresentar os top 3 Clientes

2.2.3 Requisitos de controlo

- Sr. Manuel o gerente da empresa Uminho Entregas terá permissão para visualizar toda a informação armazenada na base de dados, terá permissão para adicionar novas encomendas, clientes, veículos, funcionários, contactos e inserir na tabela 'utiliza' a matrícula do carro e o id da encomenda que foi transportada no carro. Este terá também permissão para utilização de todas as funções e procedimentos desenvolvidos.

- Utilizadores não administrativos da empresa (Funcionários). Estes têm acesso à atualização do estado da encomenda para poderem atualizar o estado para 'Entregue' logo após a entrega, à visualização das encomendas para saberem quais encomendas eles estão encarregues de entregar, e os contactos dos clientes em caso de terem de comunicar com os mesmos.

2.3 Análise e validação geral dos requisitos

Prevemos um bom funcionamento da base de dados face aos requisitos levantados, visamos todas as necessidades tanto nível de armazenamento da informação como também a nível de consulta e gestão da mesma. Da mesma forma a integridade, proteção e encapsulamento dos dados também é devidamente endereçada, com os diferentes usuários e formas de acesso á mesma.

3. Modelação Conceptual

Nos tópicos anteriores, estivemos a realizar a análise dos requisitos, agora chega a parte em que apresentamos a modelação conceptual, escrita em notação CHEN, utilizando a ferramenta brModelo. Isto com a finalidade de termos uma identificação clara e precisa sobre diversos componentes pertinentes a esta mesma, tais como: as entidades, os relacionamentos entre estas e os atributos e chaves, correspondentes a cada entidade.

3.1 Apresentação da abordagem de modelação realizada

Primeiramente, começamos por proceder para a fase de tradução dos requisitos nos seus respetivos componentes atómicos, como por exemplo, as entidades. Em seguida, analisamos associações entre as entidades que utilizamos, e ainda os seus relacionamentos. Por fim, associamos os atributos relativos a estas entidades.

3.2 Identificação e caracterização das entidades

Nome da Entidade	Descrição	Ocorrências
Encomenda	Entidade que acaba por guardar todos os dados necessários para que a entrega se realize na perfeição, e sem esquecimento de qualquer informação, como tínhamos como objetivo inicial.	Cada Encomenda tem um único Funcionário associado, assim como uma só fatura e um só cliente. Temos ainda que, cada Encomenda utiliza 1 ou mais veículos associados.
Funcionário	Termo utilizado para representar quem torna possível as entregas, da UminhoEntregas, se realizarem.	Cada Funcionário entrega zero ou mais Encomendas.
Veículo	Meio de transporte, que auxilia a tarefa de entrega das encomendas.	Cada Veículo utiliza zero ou mais Encomendas.
Cliente	Termo utilizado para representar quem usa a UminhoEntregas para realizar as suas entregas.	Cada Cliente possui zero ou mais Encomendas associadas.

Tabela 1 – Dicionário de dados de entidades

3.3 Identificação e caracterização dos relacionamentos

Em seguida, apresentaremos os relacionamentos presentes, neste modelo conceptual:

Entidade 1	Multiplicidade 1	Relacionamento	Entidade 2	Multiplicidade 2
Funcionário	1-1	Entrega	Encomenda	0-n
Cliente	1-1	Possui	Encomenda	0-n
Encomenda	0-n	Utiliza	Veículo	1-n

Tabela 2 - Dicionário sobre os dados dos relacionamentos

3.4 Identificação e caracterização da associação dos atributos com as entidades e relacionamentos.

Nesta secção são apresentados os atributos presentes no modelo conceptual:

Entidade	Atributo	Descrição	Tipo de dados	Null	Multivalor
Encomenda	Id	Identifica a encomenda que está a ser tratada	INT	Não	Não
	Estado	Indicador de como se encontra a encomenda a ser tratada	VARCHAR(15)	Não	Não
	DataEncomenda	A data em que a encomenda foi efetuada	DATE	Não	Não
	DataEntrega	A data em que a encomenda foi entregue	DATE	Sim	Não
	Taxa	Taxa da encomenda, isto como forma de obter o lucro da UminhoEntregas	FLOAT	Não	Não
	Peso	Identifica o peso da encomenda	FLOAT	Não	Não
	Conteudo	Descrição do que a encomenda contém	VARCHAR(45)	Sim	Não
	LocalEntrega	Indicação de onde a encomenda tem de ser entregue	VARCHAR(45)	Não	Não
	TipoPagamento	Forma de como a encomenda vai ser paga	VARCHAR(20)	Não	Não
	Urgente	Identifica a urgência ou não da encomenda	BIT(1)	Não	Não
	DataFatura	A data relativa ao momento de emissão da fatura	DATE	Não	Não
	ValorFatura	Montante total, relativo ao total que da encomenda	FLOAT	Não	Não

Entidade	Atributo	Descrição	Tipo de dados	Null	Multivalor
Funcionário	Id	Identifica o funcionário da UminhoEntregas	INT	Não	Não
	Nome	Nome do funcionário em questão	VARCHAR(50)	Não	Não
	Telemovel	Número de contacto do funcionário	CHAR(9)	Sim	Não
Cliente	Nif	Número que identifica o cliente da UminhoEntregas	CHAR(9)	Não	Não
	Nome	Nome do cliente a tratar	VARCHAR(50)	Não	Não
	Email	Contacto, via eletrónica, do cliente	VARCHAR(45)	Sim	Não
	Cod.postal	Número que identifica a área geográfica, da morada do cliente	CHAR(8)	Não	Não
	Localidade	Identifica a localidade de residência do cliente	VARCHAR(45)	Não	Não
	Rua	Identifica a rua de residência do cliente	VARCHAR(45)	Não	Não
	Contacto	Contacto, via telefónica, seja telefone ou telemóvel	INT	Não	SIM

Entidade	Atributo	Descrição	Tipo de dados	Null	Multivalor
Veículo	Matricula	Identifica unicamente o veículo, através da matrícula	CHAR(8)	Não	Não
	Marca	A marca do veículo em questão	VARCHAR(20)	Não	Não
	Modelo	O modelo do veículo em questão	VARCHAR(20)	Não	Não
	AnoFabrico	Ano de fabrico do veículo em questão	YEAR(4)	Não	Não
	Observacoes	Observações que se devam a ter em conta, no veículo em questão	VARCHAR(150)	Sim	Não

Tabela 3 - Dicionário sobre os atributos das Entidades

3.5 Detalhe ou generalização de entidades

Nesta secção irá se apresentar uma descrição de cada uma das entidades identificadas pelo grupo de trabalho.

- **Encomenda:** Uma vez que a empresa UminhoEntregas, necessita de um sistema de gestão de encomendas/entregas, naturalmente terá de existir a entidade de Encomendas, uma vez que esta empresa alberga um vasto leque destas.
- **Funcionário:** Achamos trivial a existência desta entidade, visto que o Sr. Manuel precisa de ter controlo sobre os funcionários que tem e para que possa ter noção de que funcionários efetuaram certa encomenda.
- **Cliente:** Esta entidade é naturalmente necessária, porque o Sr. Manuel não teria necessidade de pedir a estes 4 estudantes para o ajudarem, se

não fosse tão alta a adesão a estas entregas por parte da UminhoEntregas e posto isto, é porque há vários utilizadores e portanto esta entidade é imprescindível.

- **Veículo:** Os veículos são fundamentais para o processamento de entrega das encomendas a realizar, a necessidade de esta entidade estar no sistema é trivial. Cada veículo irá possuir a marca, o modelo, o ano de fabrico, um campo opcional, para conter observações a ter em conta no veículo em questão e ainda a sua matrícula, que funciona como identificador único da entidade.

3.6 Apresentação e explicação do diagrama ER

Após termos analisado os requisitos e a respetiva tradução nas entidades e dos relacionamentos entre estas, o modelo conceptual foi elaborado, e então elaborou-se o seguinte modelo:

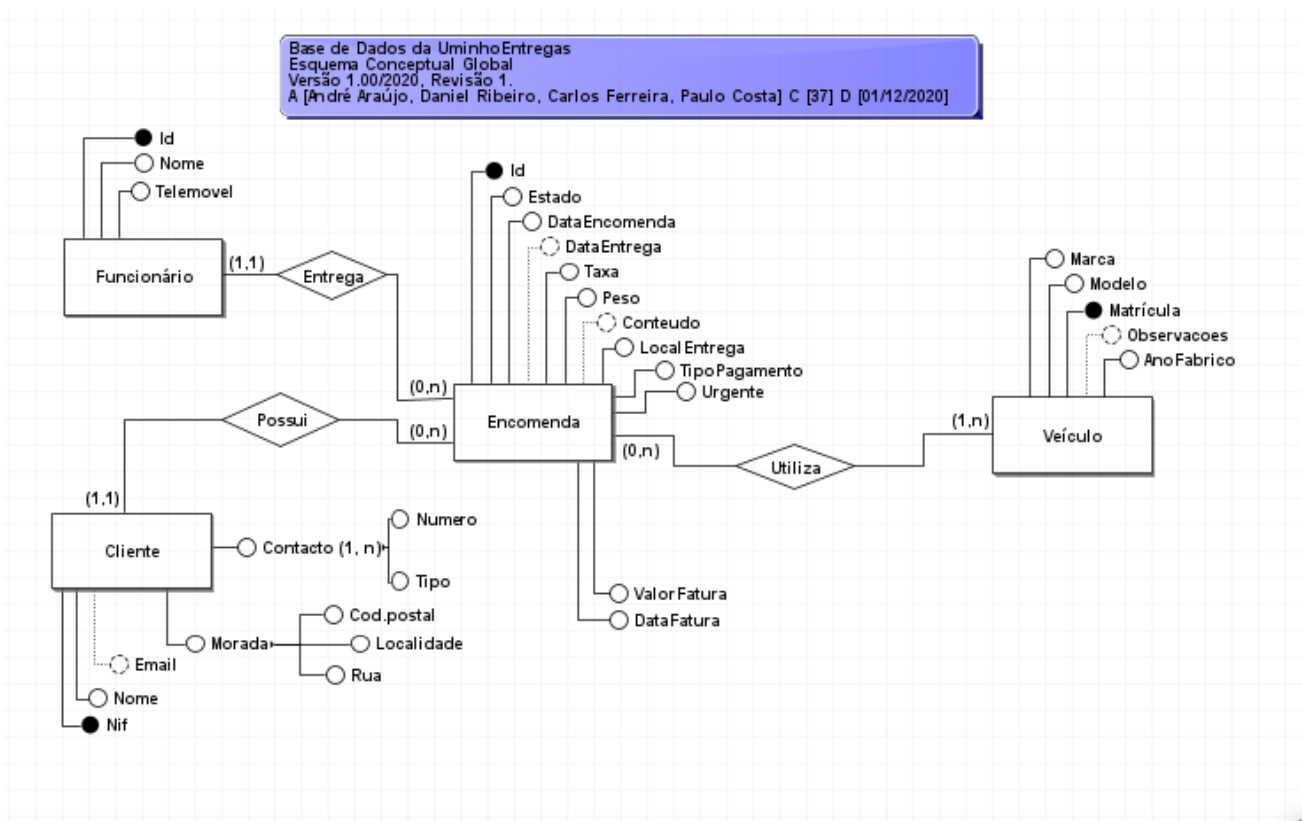


Figura 1 - Modelo Conceptual da UminhoEntregas

Utilizamos como ferramenta de construção o programa brModelo. As entidades são identificadas pelos retângulos, as relações pelos losângulos e os atributos estão associados a cada entidade e são representados pelos círculos, os círculos a tracejado correspondem aos atributos que são opcionais, isto é, que podem ser nulos. A multiplicidade de cada relacionamento está também identificada à entrada de cada entidade. Posto isto, a identificação de todos os componentes fulcrais do sistema desenvolvido é de fácil compreensão.

Os atributos e relações já foram discutidos nos pontos anteriores, contudo, de uma maneira geral o sistema de entregas da UminhoEntregas, irá possuir as entidades de funcionários, de clientes, de encomendas e ainda dos veículos (no modelo aparecem como Funcionário, Cliente, Encomenda e Veículo). Cada uma das entidades que foram referidas, possuem um identificador único para efeitos de gestão e de procura, e até de diferenciação, os restantes atributos são específicos e únicos a cada entidade, isto conforme o papel que têm nesta empresa de entregas.

Relativamente às relações da UminhoEntregas, um funcionário irá entregar zero ou mais encomendas, isto é necessário representar através de uma relação, pois a organização de muitas encomendas exige um bom planeamento de entregas. Temos ainda que, um cliente irá possuir zero ou mais encomendas, desta vez é necessário ter esta relação, para maior controlo sobre as encomendas realizadas por certos clientes. Finalmente, temos que uma encomenda utiliza um ou mais veículos, pois para efetuar a entrega é necessário ter um meio de transporte para isso, poderá ser necessário mais viaturas se de facto a encomenda for muito grande, ou se a encomenda por algum motivo tiver de ser dividida.

3.7 Validação do modelo de dados produzido

Os 4 estudantes, depois de terminada a fase de modelação conceptual, apresentaram ao Sr. Manuel o modelo realizado. O modelo reunia todos os requisitos proporcionando as funcionalidades e capacidade de gestão e de consulta que este pretendia, e como tal foi devidamente aprovado.

4. Modelação Lógica

Neste capítulo, encontra-se descrito o processo de conversão do modelo conceptual para o modelo lógico seguindo os requisitos da base de dados descritos nos capítulos anteriores.

4.1 Construção e validação do modelo de dados lógico

Nesta parte, serão apresentados os passos necessários para a derivação do modelo logico a partir do modelo conceptual.

4.1.1 Derivação das relações

Para derivar as relações para o modelo logico, temos de olhar para as entidades identificadas no modelo conceptual e as suas relações, a partir delas resultam as seguintes tabelas no modelo lógico:

- Funcionários (**id**, Nome, Telemovel):
 - O id é a chave primária e é representada por um INT;
 - O Nome é representados por um VARCHAR (50);
 - E o Telemovel é representado por CHAR (9).
- Cliente (**NIF**, Nome, Email, Cod_Postal, Localidade, Rua):
 - O NIF é a chave primária e é representada por um CHAR (9);
 - O Nome é representados por um VARCHAR (50);
 - Email é um VARCHAR (45);
 - Cod_Postal é um CHAR (8);
 - Localidade e Rua são VARCHAR (45).
- Contacto (**Numero**, Tipo, NIF_Cliente):
 - O Numero é a chave primária e é representada por CHAR (9);
 - O Tipo é um VARCHAR (45);
 - E o NIF_Cliente é uma chave estrangeira resultante da relação com Cliente.

- Encomendas (**id**, Estado, DataEncomenda, DataEntrega, Taxa, Peso, Conteudo, LocalEntrega, TipoPagamento, Urgente, ValorFatura, DataFatura, funcionarios_id, cliente_id)
 - O id é a chave primária e é representada por um INT;
 - O Estado é representado por um VARCHAR (15);
 - O Conteudo e o LocalEntrega são representados por um VARCHAR (45);
 - O TipoPagamento é representado por um VARCHAR (20);
 - A DataEncomenda e DataEntrega por DATE;
 - A Taxa e o Peso são um Float;
 - Urgente é um BIT (1);
 - A Data é representada por DATE;
 - O valor é representado por um Float.
 - funcionarios_id, fatura_id, cliente_id são chaves estrangeiras, resultantes das relações com as entidades Funcionarios, Fatura e Cliente, respetivamente.
- Veiculo (**Matricula**, Marca, Modelo, AnoFabrico, Observacoes):
 - O Matricula é a chave primária e é representada por um CHAR (8);
 - Marca e Modelo são representados por um VARCHAR (20);
 - AnoFabrico é um YEAR (4);
 - E Observacoes é um VARCHAR(150).
- Utiliza (id_encomenda, Matricula_Veiculo):
 - id_encomenda e Matricula_Veiculo são chaves primárias e chaves estrangeiras, resultantes das relações com as entidades Encomendas e Veiculo, respetivamente.

4.1.2 Derivação dos relacionamentos

Realizada a derivação das relações para as tabelas do modelo logico, derivamos os relacionamentos.

- Relacionamentos de um para muitos (1:N)
 - Funcionarios para Encomendas:
Funcionarios (id [PK]) -> Encomenda (id [PK], funcionario_id [FK]);
 - Clientes para Encomendas:
Clientes (NIF [PK]) -> Encomenda (id [PK], cliente_id [FK]);
 - Contactos para Clientes:
Clientes (NIF [PK]) -> Contacto (Numero [PK], cliente_id [FK]);
- Relacionamentos de muitos para muitos (N:N)
 - Encomendas para Veiculo:
Tendo sido criada para esta relação uma tabela auxiliar Utiliza transformando em duas relações (1:N):

Encomenda (id [PK]) -> Utiliza (id_Encomenda [FK])

Veiculo (Matricula [PK]) -> Utiliza (Matricula_Veiculo [FK])

4.1.3 Analise das restrições de integridade

Nas restrições acima descritas são essenciais, garantindo a consistência e a exatidão dos dados no sistema de base de dados relacional. Assegurando que os dados representam a realidade modelada, respeitando certas restrições:

- **Integridade de entidade**

Indica que cada tabela deve ter uma chave primária e que a coluna ou as colunas escolhidas para serem a chave primária devem ser únicas e não nulas;

- **Integridade referencial**

Estabelece que qualquer valor de chave estrangeira pode ser apenas em um de dois estados: normalmente o valor de chave primária de alguma outra tabela ou, NULL.

- **Integridade de domínio**

Especifica que as colunas de uma tabela em um banco de dados relacional devem ser declaradas em um domínio definido. Esta restrição esta garantida na análise dos dicionários de dados.

Ora seguindo as restrições de integridade acima descrita cada tabela deve ter uma chave primária com valores únicos e não podem ser NULL. E pode conter chaves estrangeiras com valor da chave primária da tabela a que se refere ou NULL caso não haja relação entre os objetos representados no banco de dados.

- Funcionários (**id**, Nome, Telemovel):

Chave Primaria: id (NOT NULL, Unique)

- Cliente (**NIF**, Nome, Email, Cod_Postal, Localidade, Rua)

Chave Primaria: NIF (NOT NULL, Unique)

- Contacto (**Numero**, Tipo, NIF_Cliente)

Chave Primaria: id (NOT NULL, Unique)

Chave Estrangeira: NIF_Cliente (NOT NULL) referente a Cliente(NIF)

- Encomendas (**id**, Estado, DataEncomenda, DataEntrega, Taxa, Peso, Conteudo, LocalEntrega, TipoPagamento, Urgente, ValorFatura, DataFatura, funcionarios_id, cliente_id)

Chave Primaria: id (NOT NULL, Unique)

Chave Estrangeira: cliente_id (NOT NULL) referente a Cliente (NIF)

funcionarios_id (NOT NULL) referente a Funcionarios (id)

- Veiculo (**Matricula**, Marca, Modelo, AnoFabrico, Observacoes):

Chave Primaria: Matricula (NOT NULL, Unique)

- Utiliza (id_encomenda, Matricula_Veiculo):

Chave Primaria: id_encomenda (NOT NULL)

Matricula_Veiculo (NOT NULL)

Chave Estrangeira: id_encomenda (NOT NULL) referente a Encomenda (id)

Matricula_Veiculo (NOT NULL) referente a Veiculo (Matricula)

Com a análise do comportamento de cada chave, verificamos que o modelo lógico cumpre todas as restrições de integridade.

4.2 Desenho do modelo lógico

Para a elaboração do modelo lógico utilizamos a ferramenta MySQL Workbench. Onde convertimos as entidades do modelo conceptual para as respetivas tabelas, com os seus atributos e os tipos representados no dicionário de dados. Onde também estabelecemos os relacionamentos entre as diferentes entidades, de forma a interligar o sistema de base de dados.

Donde resultou o seguinte modelo lógico.

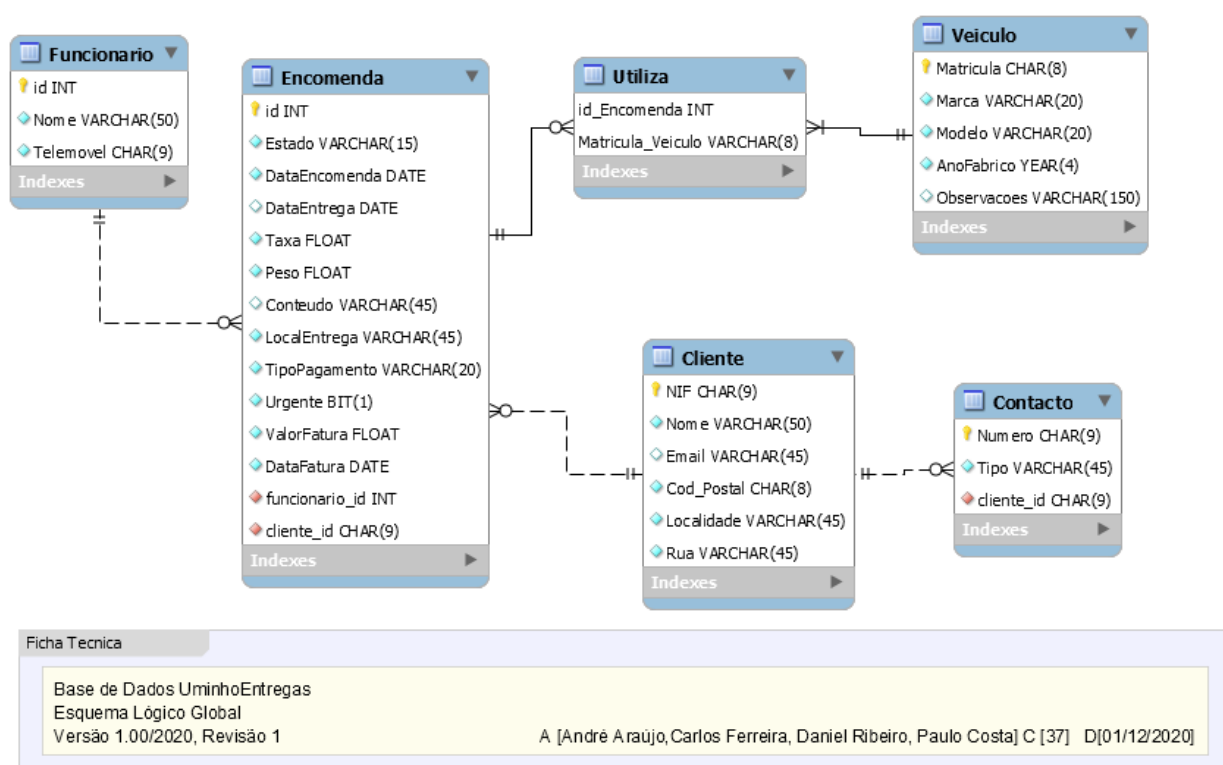


Figura 2- Modelo Lógico da UminhoEntregas

4.3 Validação do modelo através da normalização

A normalização de tabelas tem por objetivo principal resolver problemas de atualização de bases de dados, minimizando redundâncias. Ele pode ser visto como o processo no qual são eliminados esquemas de relações (tabelas) não satisfatórios, decompondo-os, através da separação de seus atributos em esquemas de relações menos complexas, mas que satisfaçam as propriedades desejadas.

- Respeitamos a 1ª Forma Normal pois todos os nossos atributos só podem ter valores atômicos
- Respeitamos a 2ª Forma Normal pois todos os nossos atributos não chave são totalmente dependentes da chave primária
- Respeitamos a 3ª Forma Normal pois todos os nossos atributos não chave são independentes entre si

Visto que seguimos todas as regras ao longo deste projeto, todas as formas normais se verificam, assim sendo a nossa base de dados está normalizada e de forma correta.

4.4 Validação do modelo com interrogações do utilizador

Bem, depois de termos o modelo conceptual realizado, justificado e devidamente validado pelo Sr. Manuel, fizemos a sua conversão para o respetivo modelo lógico. Isto, realizado com um conjunto de regras e normas necessárias para termos este modelo realizado, mas válido e adequado às necessidades de quem o utiliza, ou seja, por parte desta empresa UminhoEntregas.

Este modelo disponibiliza todas as funcionalidades que o Sr. Manuel necessita, isto para que o problema de não conseguir controlar tantas entregas, fique resolvido. Cumpre todos os requisitos e imposições estabelecidos no início do projeto.

4.5 Revisão do modelo lógico produzido

Depois de o grupo se ter reunido com o Sr. Manuel, este acabou por confirmar que o modelo apresentava os requisitos pretendidos. Desta forma, a empresa UminhoEntregas, acabaria por aceitar e validar o modelo lógico por parte destes 4 estudantes.

5. Implementação Física

5.1 Seleção do sistema de gestão de bases de dados

Para a realização do sistema de gestão de bases de dados escolhemos o MySQL, visto que é a ferramenta que estamos familiarizados, sendo a esta a usada para as aulas.

A ferramenta é bastante prática, a partir do modelo lógico, podemos gerar as instruções de criação da base de dados proporcionando assim grande eficiência no trabalho.

5.2 Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL

A partir do esquema lógico do capítulo anterior, podemos facilmente traduzir para o respetivo esquema físico usando a ferramenta do *Foward Engineering*, do MySQL sendo esta ferramenta de extrema utilidade e fácil utilização, assim o script de criação gerado foi o seguinte:

```
CREATE SCHEMA IF NOT EXISTS `UminhoEntregas` DEFAULT CHARACTER SET utf8 ;  
USE `UminhoEntregas` ;
```

Figura 3 - Criação do espaço de trabalho

```
} CREATE TABLE IF NOT EXISTS `UminhoEntregas`.`Funcionario` (  
  `id` INT NOT NULL,  
  `Nome` VARCHAR(50) NOT NULL,  
  `Telemove1` CHAR(9) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE)  
ENGINE = InnoDB;
```

Figura 4 - Criação da tabela Funcionario

```
CREATE TABLE IF NOT EXISTS `UminhoEntregas`.`Cliente` (  
  `NIF` CHAR(9) NOT NULL,  
  `Nome` VARCHAR(50) NOT NULL,  
  `Email` VARCHAR(45) NULL,  
  `Cod_Postal` CHAR(8) NOT NULL,  
  `Localidade` VARCHAR(45) NOT NULL,  
  `Rua` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`NIF`),  
  UNIQUE INDEX `NIF_UNIQUE` (`NIF` ASC) VISIBLE)  
ENGINE = InnoDB;
```

Figura 5 - Criação da tabela Cliente


```

CREATE TABLE IF NOT EXISTS `UminhoEntregas`.`Encomenda` (
  `id` INT NOT NULL,
  `Estado` VARCHAR(15) NOT NULL,
  `DataEncomenda` DATE NOT NULL,
  `DataEntrega` DATE NULL,
  `Taxa` FLOAT NOT NULL,
  `Peso` FLOAT NOT NULL,
  `Conteudo` VARCHAR(45) NULL,
  `LocalEntrega` VARCHAR(45) NOT NULL,
  `TipoPagamento` VARCHAR(20) NOT NULL,
  `Urgente` BIT(1) NOT NULL,
  `ValorFatura` FLOAT NOT NULL,
  `DataFatura` DATE NOT NULL,
  `funcionario_id` INT NOT NULL,
  `cliente_id` CHAR(9) NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_Encomendas_Funcionario1_idx` (`funcionario_id` ASC) VISIBLE,
  INDEX `fk_Encomendas_Cliente1_idx` (`cliente_id` ASC) VISIBLE,
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE,
  CONSTRAINT `fk_Encomendas_Funcionario1`
    FOREIGN KEY (`funcionario_id`)
    REFERENCES `UminhoEntregas`.`Funcionario` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Encomendas_Cliente1`
    FOREIGN KEY (`cliente_id`)
    REFERENCES `UminhoEntregas`.`Cliente` (`NIF`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 6 - Criação da tabela Encomenda

```

CREATE TABLE IF NOT EXISTS `UminhoEntregas`.`Veiculo` (
  `Matricula` CHAR(8) NOT NULL,
  `Marca` VARCHAR(20) NOT NULL,
  `Modelo` VARCHAR(20) NOT NULL,
  `AnoFabricao` YEAR(4) NOT NULL,
  `Observacoes` VARCHAR(150) NULL,
  PRIMARY KEY (`Matricula`),
  UNIQUE INDEX `Matricula_UNIQUE` (`Matricula` ASC) VISIBLE)
ENGINE = InnoDB;

```

Figura 7 - Criação da tabela Veiculo

```

CREATE TABLE IF NOT EXISTS `UminhoEntregas`.`Contacto` (
  `Numero` CHAR(9) NOT NULL,
  `Tipo` VARCHAR(45) NOT NULL,
  `cliente_id` CHAR(9) NOT NULL,
  PRIMARY KEY (`Numero`),
  INDEX `fk_Contacto_Cliente1_idx` (`cliente_id` ASC) VISIBLE,
  UNIQUE INDEX `Numero_UNIQUE` (`Numero` ASC) VISIBLE,
  CONSTRAINT `fk_Contacto_Cliente1`
    FOREIGN KEY (`cliente_id`)
      REFERENCES `UminhoEntregas`.`Cliente` (`NIF`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 8 - Criação da tabela Contacto

```

CREATE TABLE IF NOT EXISTS `UminhoEntregas`.`Utiliza` (
  `id_Encomenda` INT NOT NULL,
  `Matricula_Veiculo` VARCHAR(8) NOT NULL,
  INDEX `fk_Utiliza_Veiculo1_idx` (`Matricula_Veiculo` ASC) VISIBLE,
  PRIMARY KEY (`id_Encomenda`, `Matricula_Veiculo`),
  CONSTRAINT `fk_Utiliza_Encomendas1`
    FOREIGN KEY (`id_Encomenda`)
    REFERENCES `UminhoEntregas`.`Encomenda` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Utiliza_Veiculo1`
    FOREIGN KEY (`Matricula_Veiculo`)
    REFERENCES `UminhoEntregas`.`Veiculo` (`Matricula`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 9 - Criação da tabela Utiliza

Para facilitar o povoamento das tabelas criamos funções de inserção:

```

DELIMITER $$
CREATE FUNCTION calculaValor(peso FLOAT, taxa FLOAT, urgencia BIT ) RETURNS float
  DETERMINISTIC NO SQL READS SQL DATA
  BEGIN
    DECLARE i FLOAT DEFAULT peso * taxa;
    RETURN IF(urgencia, i * 1.20, i);
END$$
DELIMITER ;

```

Figura 10 – Função calculaValor para calcular o valor a pagar na fatura

```

DELIMITER $$
CREATE PROCEDURE `insereCliente` (IN id CHAR(9) , IN nome VARCHAR(50) , IN email CHAR(45),
                                IN codp CHAR(8) , IN loc VARCHAR(45) , IN rua VARCHAR(45))
BEGIN
    INSERT INTO cliente (NIF, Nome, Email, Cod_Postal, Localidade, Rua)
    VALUES (id , nome, email, codp , loc , rua);
END $$

```

Figura 11 – Função de inserção de um novo Cliente

```

DELIMITER $$
CREATE PROCEDURE `insereContacto` (IN num CHAR(9) , IN tipo VARCHAR(45) , IN cliente_id CHAR(9))
BEGIN
    INSERT INTO contacto ( Numero , Tipo , cliente_id)
    VALUES (num , tipo, cliente_id);
END $$

```

Figura 12 – Função de inserção de um novo Contacto

```

DELIMITER $$
CREATE PROCEDURE `insereEncomenda` (IN id INT,IN Estado VARCHAR(15),IN DataEncomenda DATE,
IN DataEntrega DATE,IN Taxa FLOAT,IN Peso FLOAT,IN Conteudo VARCHAR(45),
IN LocalEntrega VARCHAR(45), IN TipoPagamento VARCHAR(20),IN Urgente BIT(1),
IN ValorFatura FLOAT,IN DataFatura DATE,IN funcionario_id INT, IN cliente_id CHAR(9))
BEGIN
    INSERT INTO encomenda (id , Estado , DataEncomenda , DataEntrega , Taxa , Peso , Conteudo ,
    LocalEntrega , TipoPagamento , Urgente , ValorFatura , DataFatura , funcionario_id , cliente_id)
    VALUES (id , Estado , DataEncomenda , DataEntrega , Taxa , Peso , Conteudo , LocalEntrega ,
    TipoPagamento , Urgente , ValorFatura , DataFatura , funcionario_id , cliente_id);
END $$

```

Figura 13 – Função de inserção de uma nova Encomenda

```

DELIMITER $$
CREATE PROCEDURE `insereFuncionario` (IN id INT , IN nome VARCHAR(50) , IN telemovel CHAR(9))
BEGIN
    INSERT INTO funcionario ( id , Nome , Telemovel)
    VALUES ( id , nome , telemovel);
END $$

```

Figura 14 – Função de inserção de um novo Funcionário

```

DELIMITER $$
CREATE PROCEDURE `insereUtiliza` (IN id_enc INT , IN matricula VARCHAR(8))
BEGIN
INSERT INTO utiliza ( id_Encomenda , Matricula_Veiculo )
VALUES (id_enc , matricula);
END $$

```

Figura 15 – Função de inserção na tabela utiliza

```

DELIMITER $$
CREATE PROCEDURE `insereVeiculo` (IN matricula CHAR(8) , IN marca VARCHAR(20) , IN modelo VARCHAR(20) , IN ano YEAR , IN obs VARCHAR(150))
BEGIN
INSERT INTO veiculo ( Matricula, Marca , Modelo , AnoFabrico , Observacoes)
VALUES ( matricula, marca , modelo , ano , obs);
END $$

```

Figura 16 – Função de inserção de um novo veículo

Desta forma as tabelas foram povoadas através do seguinte conjunto de scripts:

```

call insereCliente('123456789' , 'Luís Marques', 'luismarques@gmail.com' , '4750-123' , 'Braga' , 'Rua das Pedras');
call insereCliente('123456790' , 'Paulo Costa', 'paulocostinha@gmail.com' , '4745-247' , 'Guimarães' , 'Rua S.Tiago Cadoso');
call insereCliente('123456791' , 'Daniel Ribeiro', 'danydimitri@gmail.com' , '4745-765' , 'Barcelos' , 'Rua da Cantoria');
call insereCliente('123456792' , 'André Araújo', null , '4745-765' , 'Barcelos' , 'Rua da Cantoria');
call insereCliente('123456793' , 'Carlos Ferreira' , null , '4745-767' , 'Barcelos' , 'Rua Barcelense');
call insereCliente('123456794' , 'Maria Teixeira', 'mariajoao@hotmail.com' , '4843-432' , 'Famalicão' , 'Rua do Estádio');
call insereCliente('123456795' , 'Ana Silva', 'anasilivia@gmail.com' , '4785-247' , 'Fafe' , 'Rua Aqui Não Fanfa');
call insereCliente('123456796' , 'Castro Diogo', 'dioguinho2008@gmail.com' , '4745-769' , 'Póvoa de Varzim' , 'Rua da Ventosa');
call insereCliente('123456797' , 'Célia Cardoso', null , '4575-765' , 'Amarante' , 'Rua do D. Infante Amaro');
call insereCliente('123456798' , 'Paula Ribeiro' , null , '4743-767' , 'Braga' , 'Rua D. Diogo');

```

Figura 17 – Povoar a tabela Clientes

```

call insereFuncionario(1,'Jose Carlos','933000123');
call insereFuncionario(2,'Rafael Costa','912456870');
call insereFuncionario(3,'Ana Castro','963933666');
call insereFuncionario(4,'Joana Gomes','923123555');
call insereFuncionario(5,'Tiago Silva','934512000');
call insereFuncionario(6,'Filipe Faria','918309377');

```

Figura 18 – Povoar a tabela Funcionários

```

call insereContacto ('912345678','Telemovel','123456789');
call insereContacto ('253441098','Telefone','123456790');
call insereContacto ('253679765','Telefone','123456791');
call insereContacto ('929292929','Telemovel','123456791');
call insereContacto ('912305463','Telemovel','123456792');
call insereContacto ('253111536','Telefone','123456792');
call insereContacto ('253444555','Telefone','123456793');
call insereContacto ('911441111','Telemovel','123456794');
call insereContacto ('933366633','Telemovel','123456795');
call insereContacto ('253854123','Telefone','123456796');
call insereContacto ('253667302','Telefone','123456797');
call insereContacto ('912345645','Telemovel','123456798');

```

Figura 19 – Povoar a tabela Contactos

```

call insereVeiculo('35-22-ZV','Opel','Corsa','2006',null);
call insereVeiculo('30-VX-53','Ford','Transit','2018',null);
call insereVeiculo('50-PD-70','Renault','Midlum','2010',null);
call insereVeiculo('ZA-50-53','Toyota','Yaris','2001','Carro não está nas melhores condições');
call insereVeiculo('66-22-AA','Audi','A4','2002','Como Novo');
call insereVeiculo('55-BP-12','Fiat','Stilo','2008',null);

```

Figura 20 – Povoar a tabela Veículos

```

call insereEncomenda( 1 , 'Pedida' , '2020/11/30' , null , 20 , 3 , null , 'Rua das Pedras' , 'Ainda não pago' , 0 , calculaValor(20,3,0) , '2020/11/30' , 1 , '123456789' );
call insereEncomenda( 2 , 'Pedida' , '2020/11/30' , null , 40 , 7 , null , 'Rua das Pedras' , 'Ainda não pago' , 0 , calculaValor(40,7,0) , '2020/11/30' , 2 , '123456789' );
call insereEncomenda( 3 , 'Pedida' , '2020/11/30' , null , 30 , 3 , null , 'Rua da Cantoria' , 'Ainda não pago' , 0 , calculaValor(30,3,0) , '2020/11/30' , 3 , '123456791' );
call insereEncomenda( 4 , 'Paga' , '2020/11/30' , null , 50 , 20 , null , 'Rua da Cantoria' , 'Online' , 0 , calculaValor(50,20,0) , '2020/11/30' , 4 , '123456792' );
call insereEncomenda( 5 , 'Paga' , '2020/11/30' , null , 20 , 8 , null , 'Rua Barcelense' , 'Online' , 0 , calculaValor(20,8,0) , '2020/11/30' , 5 , '123456793' );
call insereEncomenda( 6 , 'Paga' , '2020/11/30' , null , 25 , 10 , null , 'Rua do Estádio' , 'Online' , 0 , calculaValor(25,10,0) , '2020/11/30' , 6 , '123456794' );
call insereEncomenda( 7 , 'Paga' , '2020/11/30' , null , 10 , 30 , null , 'Rua Aqui Não Fanfa' , 'Online' , 0 , calculaValor(10,30,0) , '2020/11/30' , 7 , '123456795' );
call insereEncomenda( 8 , 'Paga' , '2020/11/30' , null , 20 , 10 , '10KG comida cão' , 'Rua da Ventosa' , 'Online' , 0 , calculaValor(20,10,0) , '2020/11/30' , 8 , '123456796' );
call insereEncomenda( 9 , 'Entregue' , '2020/11/29' , '2020/11/29' , 20 , 10 , null , 'Rua do D. Infante Amaro' , 'Cartão de Crédito' , 0 , calculaValor(20,10,0) , '2020/11/30' , 9 , '123456797' );
call insereEncomenda( 10 , 'Entregue' , '2020/11/29' , '2020/11/29' , 25 , 20 , null , 'Rua do D. Infante Amaro' , 'Dinheiro' , 0 , calculaValor(25,20,0) , '2020/11/30' , 10 , '123456798' );
call insereEncomenda( 11 , 'Entregue' , '2020/11/29' , '2020/11/29' , 25 , 7 , null , 'Rua S.Tiago Cadoso' , 'Dinheiro' , 0 , calculaValor(25,7,0) , '2020/11/29' , 11 , '123456790' );
call insereEncomenda( 12 , 'Entregue' , '2020/11/29' , '2020/11/29' , 42 , 30 , null , 'Rua D. Diogo' , 'Cartão de Crédito' , 0 , calculaValor(42,30,0) , '2020/11/29' , 12 , '123456798' );
call insereEncomenda( 13 , 'Entregue' , '2020/11/29' , '2020/11/29' , 28 , 18 , null , 'Rua da Cantoria' , 'Dinheiro' , 0 , calculaValor(28,18,0) , '2020/11/29' , 13 , '123456791' );
call insereEncomenda( 14 , 'Entregue' , '2020/11/29' , '2020/11/29' , 30 , 23 , null , 'Rua da Cantoria' , 'Cheque' , 0 , calculaValor(30,23,0) , '2020/11/29' , 14 , '123456792' );
call insereEncomenda( 15 , 'Entregue' , '2020/11/29' , '2020/11/30' , 70 , 1 , 'Colar de Ouro' , 'Rua Barcelense' , 'Cartão de Crédito' , 1 , calculaValor(70,1,0) , '2020/11/30' , 15 , '123456793' );
call insereEncomenda( 16 , 'Entregue' , '2020/11/29' , '2020/11/29' , 16 , 5 , null , 'Rua S.Tiago Cadoso' , 'Dinheiro' , 0 , calculaValor(16,5,0) , '2020/11/29' , 16 , '123456790' );
call insereEncomenda( 17 , 'Entregue' , '2020/11/29' , '2020/11/29' , 10 , 2 , null , 'Rua D. Diogo' , 'Cartão de Crédito' , 0 , calculaValor(10,2,0) , '2020/11/29' , 17 , '123456798' );
call insereEncomenda( 18 , 'Entregue' , '2020/11/28' , '2020/11/28' , 16 , 6 , 'Medicamentos' , 'Rua da Cantoria' , 'Dinheiro' , 1 , calculaValor(16,6,1) , '2020/11/28' , 18 , '123456791' );
call insereEncomenda( 19 , 'Entregue' , '2020/11/28' , '2020/11/28' , 32 , 12 , 'Medicamentos' , 'Rua da Cantoria' , 'Cheque' , 1 , calculaValor(32,12,1) , '2020/11/28' , 19 , '123456792' );
call insereEncomenda( 20 , 'Entregue' , '2020/11/28' , '2020/11/28' , 70 , 1 , 'Colar de Ouro' , 'Rua Barcelense' , 'Cartão de Crédito' , 1 , calculaValor(70,1,0) , '2020/11/28' , 20 , '123456793' );

```

Figura 21 – Povoar a tabela Encomendas com encomendas

```

call insereUtiliza(1,'35-22-ZV');
call insereUtiliza(2,'30-VX-53');
call insereUtiliza(3,'35-22-ZV');
call insereUtiliza(4,'66-22-AA');
call insereUtiliza(5,'35-22-ZV');
call insereUtiliza(6,'55-BP-12');
call insereUtiliza(7,'30-VX-53');
call insereUtiliza(8,'50-PD-70');
call insereUtiliza(9,'35-22-ZV');
call insereUtiliza(10,'50-PD-70');
call insereUtiliza(11,'66-22-AA');
call insereUtiliza(12,'30-VX-53');
call insereUtiliza(13,'ZA-50-53');
call insereUtiliza(14,'35-22-ZV');
call insereUtiliza(15,'55-BP-12');
call insereUtiliza(16,'66-22-AA');
call insereUtiliza(17,'30-VX-53');
call insereUtiliza(18,'30-VX-53');
call insereUtiliza(19,'35-22-ZV');
call insereUtiliza(20,'30-VX-53');

```

Figura 22 – Povoar a tabela utiliza

5.3 Tradução das interrogações do utilizador para SQL (alguns exemplos)

De seguida, apresentaremos as queries que visam produzir as funcionalidades da base de dados pretendidas pela empresa Uminho Entregas, funcionalidades estas que foram combinadas com a empresa durante o levantamento de requisitos.

1º Requisito: Era pretendido a apresentação de todos clientes, funcionários, veículos ou encomendas.

Para cumprimento deste requisito foram criados 4 procedimentos simples e de rápida execução.

```

DELIMITER $$
CREATE PROCEDURE `visualizarClientes` ()
) BEGIN
    SELECT * FROM cliente;
- END $$

```

Figura 23 – Procedimento para visualizar Clientes

```

DELIMITER $$
CREATE PROCEDURE `visualizarFuncionarios` ()
BEGIN
    SELECT * FROM funcionario;
END $$

```

Figura 24 – Procedimento para visualizar Funcionarios

```

DELIMITER $$
CREATE PROCEDURE `visualizarVeiculos` ()
BEGIN
    SELECT * FROM veiculo;
END $$

```

Figura 25 – Procedimento para visualizar Veiculos

```

DELIMITER $$
CREATE PROCEDURE `visualizarEncomenda` ()
BEGIN
    SELECT * FROM encomenda;
END $$

```

Figura 26 – Procedimento para visualizar Encomendas

Execução:

```

call visualizarClientes();
call visualizarFuncionarios();
call visualizarVeiculos();
call visualizarEncomenda();

```

Figura 27 – Chamada das funções

113	18:40:12	call visualizarClientes()	10 row(s) returned	0.000 sec / 0.000 sec
114	18:40:12	call visualizarFuncionarios()	6 row(s) returned	0.000 sec / 0.000 sec
115	18:40:13	call visualizarVeiculos()	6 row(s) returned	0.000 sec / 0.000 sec
116	18:40:13	call visualizarEncomenda()	20 row(s) returned	0.000 sec / 0.000 sec

Figura 28 – Funções executaram corretamente

Clientes:

	NIF	Nome	Email	Cod_Postal	Localidade	Rua
►	123456789	Luís Marques	luismarques@gmail.com	4750-123	Braga	Rua das Pedras
	123456790	Paulo Costa	paulocostinha@gmail.com	4745-247	Guimarães	Rua S.Tiago Candoso
	123456791	Daniel Ribeiro	danydimitri@gmail.com	4745-765	Barcelos	Rua da Cantoria
	123456792	André Araújo	NULL	4745-765	Barcelos	Rua da Cantoria
	123456793	Carlos Ferreira	NULL	4745-767	Barcelos	Rua Barcelense
	123456794	Maria Teixeira	mariajoao@hotmail.com	4843-432	Famalicão	Rua do Estádio
	123456795	Ana Silva	anasilvia@gmail.com	4785-247	Fafe	Rua Aqui Não Fanfa
	123456796	Castro Diogo	dioguinho2008@gmail.com	4745-769	Póvoa de Varzim	Rua da Ventosa
	123456797	Célia Cardoso	NULL	4575-765	Amarante	Rua do D. Infante Amaro
	123456798	Paula Ribeiro	NULL	4743-767	Braga	Rua D. Diogo

Figura 29 – Resultado da visualizarClientes

Funcionários:

	id	Nome	Telemovel
▶	1	Jose Carlos	933000123
	2	Rafael Costa	912456870
	3	Ana Castro	963933666
	4	Joana Gomes	923123555
	5	Tiago Silva	934512000
	6	Filipe Faria	918309377

Figura 30 – Resultado da visualizarFuncionarios

Veículos:

	Matricula	Marca	Modelo	AnoFabrico	Observacoes
▶	30-VX-53	Ford	Transit	2018	NULL
	35-22-ZV	Opel	Corsa	2006	NULL
	50-PD-70	Renault	Midlum	2010	NULL
	55-BP-12	Fiat	Stilo	2008	NULL
	66-22-AA	Audi	A4	2002	Como Novo
	ZA-50-53	Toyota	Yaris	2001	Carro não está nas melhores condições

Figura 31 – Resultado da visualizarVeiculos

Encomendas:

	id	Estado	DataEncomenda	DataEntrega	Taxa	Peso	Conteudo	LocalEntrega	TipoPagamento	Urgente	ValorFatura	DataFatura	funcionario_id	cliente_id
▶	1	Pedida	2020-11-30	NULL	20	3	NULL	Rua das Pedras	Ainda não pago	0	60	2020-11-30	1	123456789
	2	Pedida	2020-11-30	NULL	40	7	NULL	Rua das Pedras	Ainda não pago	0	280	2020-11-30	2	123456789
	3	Pedida	2020-11-30	NULL	30	3	NULL	Rua da Cantoria	Ainda não pago	0	90	2020-11-30	3	123456791
	4	Paga	2020-11-30	NULL	50	20	NULL	Rua da Cantoria	Online	0	1000	2020-11-30	4	123456792
	5	Paga	2020-11-30	NULL	20	8	NULL	Rua Barcelense	Online	0	160	2020-11-30	5	123456793
	6	Paga	2020-11-30	NULL	25	10	NULL	Rua do Estádio	Online	0	250	2020-11-30	6	123456794
	7	Paga	2020-11-30	NULL	10	30	NULL	Rua Aqui Não Fanfa	Online	0	300	2020-11-30	1	123456795
	8	Paga	2020-11-30	NULL	20	10	10KG comida cão	Rua da Ventosa	Online	0	200	2020-11-30	2	123456796
	9	Entregue	2020-11-29	2020-11-29	20	10	NULL	Rua do D. Infante Amaro	Cartão de Crédito	0	200	2020-11-30	3	123456797
	10	Entregue	2020-11-29	2020-11-29	25	20	NULL	Rua do D. Infante Amaro	Dinheiro	0	500	2020-11-30	4	123456797
	11	Entregue	2020-11-29	2020-11-29	25	7	NULL	Rua S.Tiago Candoso	Dinheiro	0	175	2020-11-29	5	123456790
	12	Entregue	2020-11-29	2020-11-29	42	30	NULL	Rua D. Diogo	Cartão de Crédito	0	1260	2020-11-29	6	123456798
	13	Entregue	2020-11-29	2020-11-29	28	18	NULL	Rua da Cantoria	Dinheiro	0	504	2020-11-29	1	123456791
	14	Entregue	2020-11-29	2020-11-29	30	23	NULL	Rua da Cantoria	Cheque	0	690	2020-11-29	2	123456792
	15	Entregue	2020-11-29	2020-11-30	70	1	Colar de Ouro	Rua Barcelense	Cartão de Crédito	1	70	2020-11-30	3	123456793
	16	Entregue	2020-11-29	2020-11-29	16	5	NULL	Rua S.Tiago Candoso	Dinheiro	0	80	2020-11-29	3	123456790
	17	Entregue	2020-11-29	2020-11-29	10	2	NULL	Rua D. Diogo	Cartão de Crédito	0	20	2020-11-29	4	123456798
	18	Entregue	2020-11-28	2020-11-28	16	6	Medicamentos	Rua da Cantoria	Dinheiro	1	115.2	2020-11-28	5	123456791
	19	Entregue	2020-11-28	2020-11-28	32	12	Medicamentos	Rua da Cantoria	Cheque	1	460.8	2020-11-28	4	123456792
	20	Entregue	2020-11-28	2020-11-28	70	1	Colar de Ouro	Rua Barcelense	Cartão de Crédito	1	70	2020-11-28	1	123456793

Figura 32 – Resultado da visualizarEncomendas

2º Requisito: Era pretendido apresentar as encomendas de um dado cliente.

A query criada receberá como input o NIF do cliente (chave primária do mesmo), e serão seleccionas todas as encomendas cujo atributo cliente_id é igual ao NIF passado como input.

```

DELIMITER $$
CREATE PROCEDURE `encomendasDeCliente` (IN nif CHAR(9))
BEGIN
    SELECT * FROM encomenda
    WHERE encomenda.cliente_id=nif;

END $$

```

Figura 33 – Query para 2º Requisito

```
call encomendasDeCliente ('123456798');
```

Figura 34 – Chamada da função

	id	Estado	DataEncomenda	DataEntrega	Taxa	Peso	Conteudo	LocalEntrega	TipoPagamento	Urgente	ValorFatura	DataFatura	funcionario_id	cliente_id
▶	12	Entregue	2020-11-29	2020-11-29	42	30	NULL	Rua D. Diogo	Cartão de Crédito	0	1260	2020-11-29	6	123456798
	17	Entregue	2020-11-29	2020-11-29	10	2	NULL	Rua D. Diogo	Cartão de Crédito	0	20	2020-11-29	4	123456798

Figura 35 – Encomendas do cliente com NIF 123456789

3º Requisito: Apresentar as encomendas entregues num dado intervalo de tempo por certo funcionário.

A query criada receberá como input o ID do funcionário (chave primária do mesmo), juntamente com as datas inicial e final do intervalo de tempo a ser estudado. Serão selecionadas todas as encomendas cujo atributo `funcionario_id` (que representa o funcionário que entregou a encomenda em causa) é igual ao ID passado como input, e cujo atributo `DataEntrega` pertença ao intervalo pretendido.

```

DELIMITER $$
CREATE PROCEDURE `encomendasEntreguesPorFuncionario` (IN idFuncionario INT , IN dataInicial DATE , IN dataFinal DATE)
BEGIN
    SELECT * FROM encomenda
    WHERE encomenda.funcionario_id=idFuncionario
    AND encomenda.Estado='Entregue'
    AND dataInicial <= encomenda.DataEntrega
    AND dataFinal >= encomenda.DataEntrega;

END $$

```

Figura 36 – Query para 3º Requisito

```
call encomendasEntreguesPorFuncionario(1 , '2020-11-28' , '2020-11-30');
```

Figura 37 – Chamada da função

	id	Estado	DataEncomenda	DataEntrega	Taxa	Peso	Conteudo	LocalEntrega	TipoPagamento	Urgente	ValorFatura	DataFatura	funcionario_id	cliente_id
▶	13	Entregue	2020-11-29	2020-11-29	28	18	NULL	Rua da Cantoria	Dinheiro	0	504	2020-11-29	1	123456791
	20	Entregue	2020-11-28	2020-11-28	70	1	Colar de Ouro	Rua Barcelense	Cartão de Crédito	1	70	2020-11-28	1	123456793

Figura 38 – Encomendas entregues pelo funcionário 1 entre dia 28 e 30 de Outubro 2020

4º Requisito: Apresentar o valor pendente de um cliente.

A query criada receberá como input o ID do cliente (chave primária do mesmo). Serão selecionados da junção das tabelas cliente e encomenda (INNER JOIN) pela chave primária da tabela cliente e chave estrangeira 'cliente_id' da tabela encomenda, o nome do cliente, NIF do cliente e o somatório dos valores das encomendas ainda não pagas, ou seja, estado da encomenda = pedida.

```
DELIMITER $$
CREATE PROCEDURE `valorPendenteCliente` (IN nif CHAR(9))
BEGIN
    SELECT cliente.NIF ,cliente.Nome,sum(encomenda.ValorFatura) AS Valor FROM cliente
    INNER JOIN encomenda
    ON cliente.NIF=encomenda.cliente_id
    WHERE cliente.NIF=nif AND encomenda.Estado='Pedida';
END $$
```

Figura 39 – Query para 4º Requisito

```
call valorPendenteCliente('123456789');
```

Figura 40 – Chamada da função

	NIF	Nome	Valor
	123456789	Luís Marques	340

Figura 41 – Valor pendente do cliente

5º Requisito: Desempenho de todos os funcionários num determinado dia.

A query criada receberá como input o dia (tipo date). Serão selecionados o id do funcionário, número de entregas (count(*)) e o somatório dos valores das entregas da tabela encomenda. Depois especificamos que apenas queremos considerar as encomendas que foram entregues no dia que foi passado como input, e finalmente agrupamos os resultados pelo 'funcionario_id'.

```

DELIMITER $$
CREATE PROCEDURE `desempenhoDosFuncionariosNoDia` (IN dia DATE)
BEGIN
    SELECT encomenda.funcionario_id , count(*) AS NrEntregas , sum(encomenda.ValorFatura) AS ValorFaturado FROM encomenda
    WHERE encomenda.DataEntrega = dia
    GROUP BY encomenda.funcionario_id
    ORDER BY 1;
END $$

```

Figura 42 – Query para 5º Requisito

```
call desempenhoDosFuncionariosNoDia('2020-11-29');
```

Figura 43 – Chamada da função

	funcionario_id	NrEntregas	ValorFaturado
▶	1	1	504
	2	1	690
	3	2	280
	4	2	520
	5	1	175
	6	1	1260

Figura 44 – Desempenho dos funcionários

6º Requisito: Veículos mais utilizados pela empresa.

Serão selecionadas a matrícula do veículo e a contagem doo número de linhas da tabela 'utiliza', depois agrupamos pela matrícula no que vai resultar a que para cada matrícula obteremos a contagem dessa na tabela, e depois ordenamos pela contagem descendente ficando assim os veículos mais utilizados no topo.

```

CREATE PROCEDURE `utilizacaoVeiculos` ()
BEGIN
    SELECT utiliza.Matricula_Veiculo , count(*) AS NrEntregas FROM utiliza
    GROUP BY utiliza.Matricula_Veiculo
    ORDER BY 2 DESC;
END $$

```

Figura 45 – Query para 6º Requisito

```
call utilizacaoVeiculos();
```

Figura 46 – Chamada da função

	Matricula_Veiculo	NrEntregas
▶	30-VX-53	6
	35-22-ZV	6
	66-22-AA	3
	50-PD-70	2
	55-BP-12	2
	ZA-50-53	1

Figura 47 – Utilização dos veículos

7º Requisito: Apresentar o top 3 clientes.

Serão selecionadas 3 colunas (nome do cliente, NIF do cliente e o somatório dos valores de todas as encomendas desse cliente), da junção das tabelas cliente e encomenda (INNER JOIN) pela chave primária da tabela cliente e chave estrangeira 'cliente_id' da tabela encomenda. Necessitamos então de agrupar pelo NIF para cada cliente ter o próprio somatório, depois ordenamos por 3 (valor do somatório) decendentemente e caso existia dois valores iguais essas linhas serão ordenadas por 2 (nome do cliente), e depois limitamos a 3 linhas.

```
DELIMITER $$
CREATE PROCEDURE `top3Clientes` ()
BEGIN
    SELECT cliente.NIF , cliente.Nome , sum(encomenda.ValorFatura) AS 'Valor Gasto' FROM cliente
    INNER JOIN encomenda
    ON cliente.NIF = encomenda.cliente_id
    GROUP BY cliente.NIF
    ORDER BY 3 DESC , 2
    LIMIT 3;
END $$
```

Figura 48 – Query para 7º Requisito

```
call top3Clientes();
```

Figura 49 – Chamada da função

	NIF	Nome	Valor Gasto
▶	123456792	André Araújo	2150.7999877929688
	123456798	Paula Ribeiro	1280
	123456791	Daniel Ribeiro	709.1999969482422

Figura 50 – Valor pendente do cliente

Requisitos de controle:

```
Use `uminhoentregas`;  
  
CREATE USER 'SrManuel'@'localhost' IDENTIFIED BY 'gerenteUminhoEntregas';  
CREATE USER 'funcionario'@'localhost' IDENTIFIED BY 'UminhoEntregas2020';  
  
GRANT SELECT,UPDATE ON uminhoentregas.* TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON PROCEDURE insereCliente TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON PROCEDURE insereFuncionario TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON PROCEDURE insereEncomenda TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON PROCEDURE insereVeiculo TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON PROCEDURE insereContacto TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON PROCEDURE insereUtiliza TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON FUNCTION calculaValor TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON PROCEDURE visualizarClientes TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON PROCEDURE visualizarFuncionarios TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON PROCEDURE visualizarEncomenda TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON PROCEDURE visualizarVeiculos TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON PROCEDURE encomendasDeCliente TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON PROCEDURE encomendasEntreguesPorFuncionario TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON PROCEDURE valorPendenteCliente TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON PROCEDURE top3Clientes TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON PROCEDURE desempenhoDosFuncionariosNoDia TO 'SrManuel'@'localhost';  
GRANT EXECUTE ON PROCEDURE utilizacaoVeiculos TO 'SrManuel'@'localhost';  
  
GRANT UPDATE (Estado) ON uminhoentregas.encomenda TO 'funcionario'@'localhost';  
GRANT EXECUTE ON PROCEDURE visualizarEncomenda TO 'funcionario'@'localhost';  
GRANT SELECT ON uminhoentregas.contacto TO 'funcionario'@'localhost';
```

Figura 51 – Requisitos de controle

5.4 Escolha, definição e caracterização de índices em SQL (alguns exemplos)

Visto que as chaves primárias e estrangeiras já são indexadas, o único índice que achamos ser útil para além destas é o Estado das Encomendas, criamos assim o índice:

```
CREATE INDEX idxEstadoEncomenda  
ON encomenda (Estado);
```

Figura 52 - Criação do índice no Estado da Encomenda

5.5 Estimativa do espaço em disco da base de dados e taxa de crescimento anual

Vamos começar por calcular o tamanho que cada registo ocupa nas tabelas

Funcionario		
Id	int	4bytes
Nome	varchar(50)	51bytes
Telemovel	char(9)	9bytes
Total:		64bytes

Tabela 4 – Cálculo do tamanho de cada registo do Funcionario

Cliente		
NIF	char(9)	4bytes
Nome	varchar(50)	51bytes
Email	varchar(45)	46bytes
Cod_Postal	Char(8)	8bytes
Localidade	varchar(45)	46bytes
Rua	Varchar(45)	46bytes
Total:		206bytes

Tabela 5 – Cálculo do tamanho de cada registo do Cliente

Encomenda		
Id	Int	4bytes
Estado	varchar(15)	16bytes
DataEncomenda	Date	3bytes
DataEntrega	Date	3bytes
Taxa	Float	4bytes
Peso	Float	4bytes
Conteudo	varchar(45)	46bytes
LocalEntrega	varchar(45)	46bytes
TipoPagamento	varchar(20)	46bytes
Urgente	Bit(1)	1byte
ValorFatura	Float	4bytes
DataFatura	Date	3bytes
Funcionario_id	Int	4bytes
Cliente_id	Int	4bytes
Total:		168bytes

Tabela 6 – Cálculo do tamanho de cada registo da Encomenda

Veiculo		
Matricula	char(8)	8bytes
Marca	varchar(20)	21bytes
Modelo	varchar(20)	21bytes
AnoFabricao	Year(4)	1bytes
Observacoes	varchar(150)	151bytes
Total:		202bytes

Tabela 7 – Cálculo do tamanho de cada registo do Veiculo

Contacto		
Numero	char(9)	8bytes
Tipo	varchar(45)	46bytes
Cliente_id	Char(9)	21bytes
Total:		75bytes

Tabela 8 – Cálculo do tamanho de cada registo do Contacto

Utiliza		
Id_Encomenda	int	4bytes
Matricula_Veiculo	char(8)	8bytes
Total:		12bytes

Tabela 9 – Cálculo do tamanho de cada registo do Utiliza

Sendo assim assumindo o número de novos funcionários e de novos veículos por mês é desprezável e que UminhoEncomendas recebe 240 encomendas por mês

Em encomendas temos: $240 \times 168 = 40320$ Bytes

Em utiliza temos: $240 \times 12 = 2880$ Bytes

E adquire 80 clientes por mês cada com media de um só contacto por cliente:

Em clientes temos: $206 \times 80 = 16480$ Bytes

Em contacto temos $64 \times 80 = 5120$ Bytes

Somando isto temos que a base de dados cresceu 64800 Bytes por mês

Assim anualmente que a base de dados cresce 777 600 Bytes, ou seja, 0.741577 Megabytes

5.6 Definição e caracterização das vistas de utilização em SQL (alguns exemplos)

Nesta secção iremos mostrar alguns exemplos de vistas que utilizamos. Vamos apresentar três exemplos, achamos que colocando estes, conseguiríamos apresentar resoluções de diferentes complexidades, neste caso, crescentemente.

Primeiramente, criamos uma tabela, que nos tornará possível aceder à informação de uma fatura, apresentando diferentes dados da encomenda relacionada a esta. Tais como: o id, o valor e data da encomenda.

```
CREATE VIEW vwInfoFatura AS  
SELECT encomenda.id , encomenda.ValorFatura , encomenda.DataFatura FROM encomenda;
```

Figura 53 - Tabela para mostrar informação de fatura

Em seguida, uma tabela para representar diferentes médias, tais como: peso, taxa e valores de encomendas.

```
CREATE VIEW vwMedias AS  
SELECT AVG(encomenda.peso) AS MediaPeso ,  
AVG(encomenda.taxa) AS MediaTaxa ,  
AVG(encomenda.ValorFatura) AS MediaValorFatura  
FROM encomenda;
```

Figura 54 - Tabela representativa de diferentes médias

Finalmente, uma tabela para mostrar encomendas que ainda não foram entregues. Esta tabela acaba por ter um grau de dificuldade maior, pois exige mais conceitos, e por isso, decidimos deixá-la para último.

```
CREATE VIEW vwEmFalta AS
SELECT encomenda.id, peso, LocalEntrega, cliente.Nome as Nome_Cliente, funcionario.nome as Nome_Funcionario
FROM encomenda
inner join cliente on cliente_id = NIF
inner join funcionario on funcionario.id = funcionario_id
WHERE DataEntrega is null
```

Figura 55 - Tabela que mostra as encomendas em falta

5.7 Revisão do sistema implementado

Finalmente, terminada a implementação física, o grupo reuniu-se de novo com o Sr. Manuel, desta vez com o intuito de completar o trabalho que ficou delineado a estes 4 estudantes. Visto que, o sistema satisfaz os requisitos que este propôs e está apto a ajudar o Sr. Manuel, nesta tarefa que se tornou complicada, o sistema foi devidamente validado.

6. Conclusões e Trabalho Futuro

Este trabalho teve um elevado grau de importância pois ajudou-nos bastante a consolidar a matéria lecionada nas aulas sobre base de dados, desde a primeira fase da análise de requisitos até á fase de modelação conceptual como de modelação lógica e finalmente a implementação física, com aplicações reais e uteis.

Apesar de nos depararmos com diversas dificuldades, o grupo esteve á altura para tais, como por exemplo os requisitos iniciais que exigiu de nós organizar quais as entidades necessárias para o projeto, pois tivemos uma altura em que de facto tivemos entidades que podiam ser substituídas por simples atributos.

Estamos assim satisfeitos com o resultado deste trabalho pois conseguimos elaborar uma base de dados passo a passo, satisfazendo todos os requisitos pedidos.

Prevemos que a expansão futura deste projeto é possível uma vez que qualquer base de dados pode ser sempre melhorada, quer seja pela adição de novas entidades ou atributos, seja por meio de aperfeiçoamento da sua estrutura.

Finalizamos este relatório com certeza de termos acrescentado conhecimentos úteis e importantes para o nosso futuro, não só como a elaboração, modelação e levantamento de requisitos, mas também de uso das ferramentas de modelação e criação da base de dados, algo que não tínhamos muita experiência prévia.

7. Referências

Connolly, T., Begg, C., Database Systems: A Practical Approach to Design, Implementation, and Management, AddisonWesley, Global Edition, 26 Sep 2014. ISBN-10: 1292061189, ISBN-13: 978-1292061184.

Carlos Cuenca Díaz, C 2004, *Tipos de dados de MySQL*, viewed 2 Dec 2020, <<http://www.criarweb.com/artigos/118.php>>.

Lista de Siglas e Acrónimos

- 1) **BD** - Base de Dados
- 2) **SQL** – Structured Query Language
- 3) **PK** – Primary Key (Chave primária)
- 4) **FK** – Foreign Key (Chave estrangeira)

Anexos

Anexo 1